String to Integer



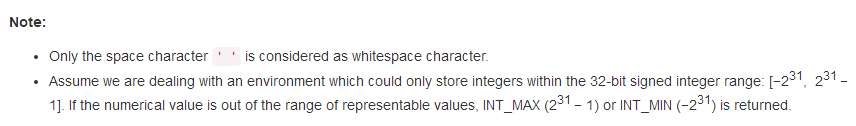
Implement atoi which converts a string to an integer.

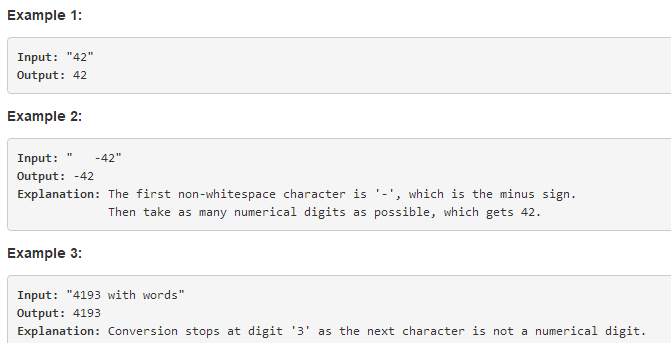
The function first discards as many whitespace characters as necessary until the first non-whitespace character is found. Then, starting from this character, takes an optional initial plus or minus sign followed by as many numerical digits as possible, and interprets them as a numerical value.

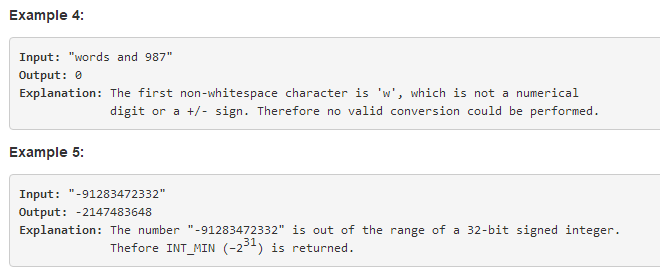
The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in str is not a valid integral number, or if no such sequence exists because either str is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned.







My Answer：

class Solution {

public int myAtoi(String str) {

if(str == null){

return 0;

}

str = str.trim();

if("".equals(str)){

return 0;

}

boolean isNegative = false;

int start = 0;

int end = str.length()-1;

while(!(('0' <= str.charAt(start) && str.charAt(start) <= '9')||str.charAt(start)=='+' || str.charAt(start) == '-')){

start ++;

if(start > end){

return 0;

}

}

if(str.length() == 1 &&(str.charAt(0)<'0'||str.charAt(0)>'9')){

return 0;

}

if(str.charAt(start) == '-'&&str.length() > 1){

isNegative = true;

str = str.substring(start+1);

}else if(str.charAt(start)=='+'&& str.length() > 1){

str = str.substring(start+1);

}

start = 0;

while(start < str.length() && str.charAt(start)=='0'){

start++;

}

str = str.substring(start);

int kkk = 0;

while(kkk < str.length() && '0' <= str.charAt(kkk) && str.charAt(kkk) <= '9'){

kkk++;

}

str = str.substring(0,kkk);

int len = str.length();

//judge the integer is or not in the range of Integer

if(len < 1 ){

return 0;

}

if(len > 10){

return isNegative ? -2147483648:2147483647;

}

if(len == 10){

if(isNegative){

return (str.compareTo("2147483648") < 0) ? Integer.parseInt("-"+str) : -2147483648;

}

return (str.compareTo("2147483647") <= 0) ? Integer.parseInt(str) : 2147483647;

}

return isNegative ? Integer.parseInt("-"+str):Integer.parseInt(str);

}

}